Modbus RTU 主站指令库

西门子在 Micro/WIN V4.0 SP5 中正式推出 Modbus RTU 主站协议库（西门子标准库指令）。

图 1. 西门子标准指令库（Micro/WIN V4.0 SP5）

注意：

1. Modbus RTU 主站指令库的功能是通过在用户程序中调用预先编好的程序功能块实现的，该库对 Port0 和 Port 1 有效。该指令库将设置通信口工作在自由口模式下。
2. Modbus RTU 主站指令库使用了一些用户中断功能，编其他程序时不能在用户程序中禁止中断。
3. Modbus RTU 主站库对 CPU 的版本有要求。CPU 的版本必须为 2.00 或者 2.01（即订货号为 6ES721*－***23-0BA*），1.22 版本之前（包括 1.22 版本）的 S7-200 CPU 不支持。

使用 Modbus RTU 主站指令库，可以读写 Modbus RTU 从站的数字量、模拟量 I/O 以及保持寄存器。

要使用 Modbus RTU 主站指令库，须遵循下列步骤：

1. 安装西门子标准指令库
2. 按照要求编写用户程序调用 Modubs RTU 主站指令库

Modbus RTU 主站功能编程

1. 调用 Modbus RTU 主站初始化和控制子程序

使用 SM0.0 调用 MBUS_CTRL 完成主站的初始化，并启动其功能控制：

图 2. 用 SM0.0 调用 Modbus RTU 主站初始化与控制子程序

各参数意义如下：

a. EN	使能：	必须保证每一扫描周期都被使能（使用 SM0.0）
b. Mode	模式：	为 1 时，使能 Modbus 协议功能；为 0 时恢复为系统 PPI 协议
c. Baud	波特	支持的通讯波特率为 1200，2400，4800，9600，19200，38400，

	率：	57600，115200。
d. Parity	校验：	校验方式选择 0＝无校验 1＝奇较验 2＝偶较验
e. Timeout	超时：	主站等待从站响应的时间，以毫秒为单位，典型的设置值为 1000 毫秒（1 秒），允许设置的范围为 1 - 32767。 注意： 这个值必须设置足够大以保证从站有时间响应。
f. Done	完成位：	初始化完成，此位会自动置 1。可以用该位启动 MBUS_MSG 读写操作（见例程）
g. Error		初始化错误代码（只有在 Done 位为 1 时有效）： 0＝ 无错误 1＝ 校验选择非法 2＝ 波特率选择非法 3＝ 模式选择非法

2. 调用 Modbus RTU 主站读写子程序 MBUS_MSG，发送一个 Modbus 请求；

图 3. 调用 Modbus RTU 主站读写子程序

各参数意义如下:

a. EN	使能：	同一时刻只能有一个读写功能（即 MBUS_MSG）使能 注意：建议每一个读写功能（即 MBUS_MSG）都用上一个 MBUS_MSG 指令的 Done 完成位来激活，以保证所有读写指令循环进行（见例程）。
b. First	读写请求位：	每一个新的读写请求必须使用脉冲触发
c. Slave	从站地址：	可选择的范围 1 -247
d. RW	从站地址：	0 ＝ 读， 1 ＝ 写 注意： 1. 开关量输出和保持寄存器支持读和写功能

		2. 开关量输入和模拟量输入只支持读功能
e. Addr	读写从站的数据地址：	选择读写的数据类型 00001 至 0xxxx - 开关量输出 10001 至 1xxxx - 开关量输入 30001 至 3xxxx - 模拟量输入 40001 至 4xxxx - 保持寄存器
f. Count	数据个数	通讯的数据个数（位或字的个数） 注意： Modbus 主站可读/写的最大数据量为 120 个字（是指每一个 MBUS_MSG 指令）
g. DataPtr	数据指针：	1. 如果是读指令，读回的数据放到这个数据区中 2. 如果是写指令，要写出的数据放到这个数据区中
h. Done	完成位	读写功能完成位
i. Error	错误代码：	只有在 Done 位为 1 时，错误代码才有效 0 = 无错误 1 = 响应校验错误 2 = 未用 3 = 接收超时（从站无响应） 4 = 请求参数错误（slave address, Modbus address, count,RW） 5 = Modbus/自由口未使能 6 = Modbus 正在忙于其它请求 7 = 响应错误（响应不是请求的操作） 8 = 响应 CRC 校验和错误 - 101 = 从站不支持请求的功能 102 = 从站不支持数据地址 103 = 从站不支持此种数据类型 104 = 从站设备故障 105 = 从站接受了信息，但是响应被延迟 106 = 从站忙，拒绝了该信息 107 = 从站拒绝了信息 108 = 从站存储器奇偶错误

常见的错误：

- 如果多个 MBUS_MSG 指令同时使能会造成 6 号错误
- 从站 delay 参数设的时间过长会造成主站 3 号错误
- 从站掉电或不运行，网络故障都会造成主站 3 号错误

3. 在 CPU 的 V 数据区中为库指令分配存储区（Library Memory）

Modbus Master 指令库需要一个 284 个字节的全局 V 存储区。

关于 Modbus RTU 主站协议库的补充说明

此为西门子正式推出的标准库指令说明资料。

在 Modbus RTU Master 协议和 PPI 协议之间切换：

Modbus RTU Master 协议指令库使通信口工作在自由口模式下，此时不能与 Micro/WIN 软件通信。要在切换回 PPI 协议，可以：

- 将 MBUS_CTRL 指令的 Mode 输入端设置为逻辑“0”
- 将 CPU 的允许模式选择开关置为 STOP 位置

Modbus RTU Master 协议库的执行时间：

Modbus RTU Master 协议库的 MBUS_CTRL 指令不需要很长的执行时间。MBUS_需要 1.11 ms 用于初始化，在后续的每个扫描周期中只占用 0.41 ms。

调用 MBUS_MSG 子程序会加长处理时间。大部分时间都用于 CRC 校验的计算。每读、写一个字的数据就需要 1.85 ms 扫描时间。数据最多的情况下（读、写 120 字的数据），扫描时间大概会扩增加 222ms。读操作的时间主要消耗在接收数据上；写操作的时间主要消耗在发送数据上。

Modbus 地址

通常 Modbus 地址由 5 位数字组成，包括起始的数据类型代号，以及后面的偏移地址。Modbus Master 协议库把标准的 Modbus 地址映射为所谓 Modbus 功能号，读写从站的数据。Modbus Master 协议库支持如下地址：

- 00001 - 09999：数字量输出（ 线圈）
- 10001 - 19999：数字量输入（触点）
- 30001 - 39999：输入数据寄存器（通常为模拟量输入）
- 40001 - 49999：数据保持寄存器

Modbus Master 协议库支持的功能

为了支持上述 Modbus 地址的读写，Modbus Master 协议库需要从站支持下列功能：

表 1. 需要从站支持的功能

Modbus 地址	读/写	Modbus 从站须支持的功能
00001 - 09999 数字量输出	读	功能 1
	写	功能 5：写单输出点 功能 15：写多输出点
10001 - 19999 数字量输入	读	功能 2
	写	—
30001 - 39999 输入寄存器	读	功能 4
	写	—
40001 - 49999 保持寄存器	读	功能 3
	写	功能 6：写单寄存器单元 功能 16：写多寄存器单元

Modbus 地址和 S7-200 存储区地址的映射

S7-200 通过 Modbus Master 和 Slave 协议库通信时，Modbus 地址和 S7-200 内存储区地址的映射关系都类似。

Modbus 保持寄存器地址映射举例：

S7-200 存储区字节寻址

Modbus 保持寄存器地址　S7-200 存储区字寻址

40001	12 34
40002	56 78
40003	9A BC

VW200	12 34
VW202	56 78
VW204	9A BC

VB200	12
VB201	34
VB202	56
VB203	78
VB204	9A
VB205	BC

Modbus 数字量地址映射举例：

位地址（0xxxx 和 1xxxx）数据总是以字节为单位打包读写。第一个字节中的最低有效位对应 Modbus 地址的起始地址。如下图所示：

图 4. 数字量地址映射举例

常问问题

Modbus RTU 主站库对 CPU 的版本是否有要求，为什么编译例子程序时，会遇到 4 个错误？

Modbus RTU 主站库对 CPU 的版本确实有要求，CPU 的版本必须为 2.00 或者 2.01(即订货号为 6ES721*－***23-0BA*)，1.22 版本之前（包括 1.22 版本）的 S7-200 CPU 不支持。

Modbus 指令库启动后，如何通过同一个通信端口进行 CPU 监控？

Modbus 指令库使用的是 CPU 的自由口通信功能，工作在自由口模式下的通讯口不能使用 Micro/WIN 的 PPI 编程通信监控。如果通信口都已经被占用，可以考虑：

- 加一个通信模块（如 EM 277、CP 243-1、EM 241 等）扩展出一个编程通信口
- 中止自由口模式，可以将 CPU 上的模式开关从 RUN 拨到 STOP；或者保持处于 RUN 状态，用程序停止指令库的 Modbus 模式（参见指令库应用）

如何理解 Modbus 地址与功能码的区别？

Modbus 地址与 Modbus 的功能码是两个层次的概念。

根据 Modbus 通信协议，Modbus 数据的地址使用 0xxxx、1xxxx、3xxxx 和 4xxxx 的形式，分别表示数字量输出、数字量输入、模拟量输入等数据地址。在使用 S7-200 的指令库时，Modbus 数据地址与 S7-200 的 I/O 和数据存储区地址间有特定的对应关系。

有些设备表明它支持 Modbus RTU 通信协议，但也详细提供了读写数据的详细通信帧格式，其中包括如何指定 Modbus 站的地址，需要读写数据类型、长度等等。数据帧有特定字节指出此指令读写的数据类型和地址，此字节的数据内容即所谓“功能码”，如功能 1 指定读取单个/多个数字量输出点的值。

支持 Modbus 协议的设备或软件，使用时用户直接设置或看到的应当是 Modbus 数据地址。Modbus 地址所访问的数据，是通过各种“功能”读写而来。功能码

是 Modbus 地址的底层。如果 Modbus 通信的一方提供的所谓 Modbus 协议只有功能码，则需要注意了解此功能号与 Modbus 地址间的对应关系。

如何访问大于 9999 的保持寄存器地址？

通常 Modbus 协议的保持寄存器地址范围在 40001 - 49999 之间。对于多数应用来说已经够了。但有些 Modbus 从站把地址映射到保持寄存器区的地址超过 9999 的部分。

Modbus Master 协议库支持超过 9999 的保持寄存器地址。地址范围为 400001 - 465536。只需在调用 MBUS_MSG 子程序时给 Addr 参数赋相应的值即可，如 416768。

Modubs Master 扩展地址模式仅支持保持寄存器区，不支持其他地址类型

使用 modbus 库常见问题

S7-200 的 Modbus RTU 主/从库使用简单方便。关于 Modbus 协议介绍和该库的具体使用，在此忽略，只贴出本人在实际运用中遇到的问题：

（1）：发生“未为库分配 V 存储区”错误，右键“程序块”-->库存储区，分配存储区，注意不能与其他存储区重叠。有次我把库存储区起始地址设为 VB500，随着程序的编写，用到了 VB500 之后的地址，导致读写指令不工作，错误代码具体是几，忘记了。查找了好久才发现该问题，期间走了许多弯路，很是郁闷。建议将库存储区尽量往后分配。

（2）：MBUS_CTRL（主）和 MBUS_INIT（从）的 Mode 可以切换 PPI 与 Modbus。有次在调试过程中发现不能向 PLC 下载程序（当时头蒙，居然忘了该细节）。最后才得知该口被 Modbus 占用了，将 CPU 的开关拨到 STOP 即可监控和下载程序。后来我通过一个钮子开关切换 Mode 位，以此切换 modbus 与 PPI，CPU 模式开关一直拨在 TEAM 档，很方便。（通过 Micro/Win 启动后，Micro/Win 已不能监控程序）

（3）：PORT0 支持 Modbus RTU 主站模式和从站模式；PORT1 只支持主站模式，不支持从站模式。至于 PORT1 不支持从站，个人觉得是因为一个 CPU 如被两个主站控制，那么相当于该 CPU 在 Modbus 网络存在多个主站控制，那么“令牌优先权”问题不好解决。如果非要将 PORT1 实现 Modbus 从站功能，可参考西门子提供的“Tip041b.mwp”。（见附件）

附件:**tip041b.zip**

[本地下载] [迅雷专用高速下载]大小：21.71KB 总下载量:**805** 次

（4）：Modbus 主站库使用了中断，在其他程序段中不可禁止中断。

（5）：在使用 Modbus 主站库时，由于同一时刻只能调用一个 MBUS_MSG（否则会发生错误代码 6，曾经我忽视该错误代码，同时调用 2～5 条读写指令，上位机读写正常，没有问题。后来为了程序严谨，取缔了该做法。）一般用上一个 MBUS_MSG 的 Done 完成位来触发下一条读写指令（相对于通过定时器读写，可以提高通讯效率）；对于 MBUS_MSG 的 First 位，只需导通一个周期即可完成一次读写，一般用上升沿触发，完成后要复位，方便下次触发。

（6）：使用 Modbus 主站库时，从站延时时间设置过长，或从站掉电、故障都会发生错误代码 3，即从站无响应。为了记录错误代码，可以将 Error 保存至某个链表，以便查看错误记录。具体做法是采用 ATT 填表指令和 FIFO 先进先出指令保存最新的 n 条错误代码（注：表指令操作的数据格式为 VW，Error 为 VB，另外一个字节可以用来保存该错误发生在哪条读写指令）。注意及时保存，当 done 位从 0 变为 1，error 会被再次刷新。保存完之后将 error 复位为 0，在通讯一直发生错误时，该做法可以通过 error 的变化来记录错误的次数。

（7）：在使用 Modbus 从站库时，首先要明白 HoldStart（寄存器区起始地址）与 MaxHold（VW 的个数），该数据区不能与库存储区重叠！假设 HoldStart 设为&VB100，用串口调试助手发送读寄存器指令，第一个为 VW100，第二个为 VW102，依次类推。

（8）：在调试 Modbus 从站时，上位机读取的双整数或浮点数不正确。这是由于西门子采用“高字节低地址、低字节高地址”机制。在上位机中将高低字交换后再转换为浮点数，或直接将 PLC 中需要读写的 DW 的高低字交换。

（9）：关于通讯状态监测：常用 SM0.5 累加保存至某个寄存器，在上位机监测该寄存器，如果 1S 变化一次，则表示通讯正常。

显然，通信时问题不止这些，软硬件必须都要严格要求，比如通信距离远时必须配置终端电阻等等。该

贴只为给初次使用该库的新手们提个醒。如有不对和不足的地方，不吝赐教，感激万分。

最后引用西门子的一句话：“记住联网的格言：你糊弄它，它就糊弄你！”